

CLAIMS

09/128,394
F.D. 08/03/98

What is claimed is:

- 5 *sd* 1. A method of analyzing multi-threaded programs, comprising:
determining that unsynchronized accesses to a resource of interest can be
performed by a plurality of threads;
receiving a request from a first thread to access the resource;
suspending the first thread; and
while the first thread is suspended, receiving a request from a second thread to
10 access the resource.
2. The method of claim 1, wherein the requests of the first and second threads
is to write data to the resource.
- 15 3. The method of claim 1, further comprising awakening the first thread.
4. The method of claim 1, further comprising logging for a user that the first
and second thread performed unsynchronized accesses to the resource.
- 20 5. The method of claim 1, wherein the first thread is suspended for a
predetermined time, meaning that the first thread awakens after the predetermined time
expires.
- 25 6. The method of claim 5, wherein the thread is also suspended on an event,
meaning that the event awakens the first thread.
7. The method of claim 6, wherein the second thread sends the event that
awakens the first thread.
- 30 8. The method of claim 1, wherein the resource is a memory location, region
of memory, hardware component, or peripheral device.
9. A computer program product for analyzing multi-threaded programs,
comprising:
35 computer code that determines that unsynchronized accesses to a resource of
interest can be performed by a plurality of threads;
computer code that receives a request from a first thread to access the resource;

09/128,394 F.D. 08/03/98

computer code that suspends the first thread;
computer code that while the first thread is suspended, receives a request from a
second thread to access the resource; and
a computer readable medium that stores the computer codes.

10. The computer program product of claim 9, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

11. A method of analyzing multi-threaded programs, comprising:
determining that unsynchronized accesses to a memory location can be performed by a plurality of threads;
receiving a request from a first thread to write data to the memory location;
suspending the first thread; and
while the first thread is suspended, receiving a request from a second thread to write data to the memory location.

12. The method of claim 11, further comprising awakening the first thread.

13. The method of claim 11, further comprising logging for a user that the first and second thread performed unsynchronized writes to the memory location.

14. The method of claim 11, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

15. The method of claim 14, wherein the thread is also suspended on an event, meaning that the event awakens the first thread.

16. The method of claim 15, wherein the second thread sends the event that awakens the first thread.

17. A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that receives a request from a first thread to write data to the memory location;

computer code that suspends the first thread;

computer code that while the first thread is suspended, receives a request from a second thread to write data to the memory location; and
a computer readable medium that stores the computer codes.

5 18. The computer program product of claim 17, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

10 19. A method of analyzing multi-threaded programs, comprising:
determining that unsynchronized accesses to a memory location can be performed by a plurality of threads;
receiving a request from a first thread to write data to the memory location;
suspending the first thread;
while the first thread is suspended, receiving a request from a second thread to
15 write data to the memory location;
awakening the first thread; and
logging for a user that the first and second thread performed unsynchronized writes to the memory location.

20 20. The method of claim 19, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

25 21. The method of claim 20, wherein the thread is also suspended on an event, meaning that the event awakens the first thread.

22. The method of claim 21, wherein the second thread sends the event that awakens the first thread.

30 23. A computer program product for analyzing multi-threaded programs, comprising:
computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;
computer code that receives a request from a first thread to write data to the
35 memory location;
computer code that suspends the first thread;
computer code that while the first thread is suspended, receives a request from a second thread to write data to the memory location;
computer code that awakens the first thread;

computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location; and
a computer readable medium that stores the computer codes.

5 24. The computer program product of claim 23, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

10 25. A method of analyzing multi-threaded programs, comprising:
modifying an existing multi-threaded program include computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

modifying the existing multi-threaded program to include computer code that suspends a first thread that writes data to a memory location; and

15 modifying the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location.

20 26. The method of claim 25, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

25 27. The method of claim 26, wherein the thread is also suspended on an event, meaning that the event awakens the first thread.

28. The method of claim 27, wherein the second thread sends the event that awakens the first thread.

30 29. A computer program product for analyzing multi-threaded programs, comprising:

computer code that modifies an existing multi-threaded program include computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

35 computer code that modifies the existing multi-threaded program to include computer code that suspends a first thread that writes data to a memory location;

computer code that modifies the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location; and

a computer readable medium that stores the computer codes.

30. The computer program product of claim 29, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash
5 memory, system memory, hard drive, and data signal embodied in a carrier wave.

SECRET